

# VIDA DE PROGRAMADOR

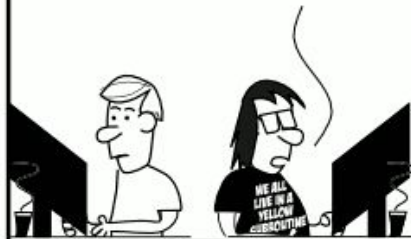
.COM.BR

/\* História real.  
Infelizmente. \*/



#1123

ESTOU TENTANDO COLOCAR  
EM ORDEM MINHAS PENDÊNCIAS...  
SEMANA PASSADA FIZ UMA "TODO  
LIST" PRA VER SE CONSIGO IR  
ELIMINANDO ELAS...



E AÍ, COMO ESTÁ  
INDO...

BOM, JÁ CONSEGUI  
ELIMINAR UMA...



## ToDo List:

- ☒ Fazer uma ToDo List
- ☐ Terminar alterações no
- ☐ Corrigir itens de acesso
- ☐ Verificar permissões de
- ☐ Arrumar layout do Vida
- ☐ Responder e-mails



Aula 06

# Array e Funções



# Posições



# Operações com Array

- Quantos elementos têm?

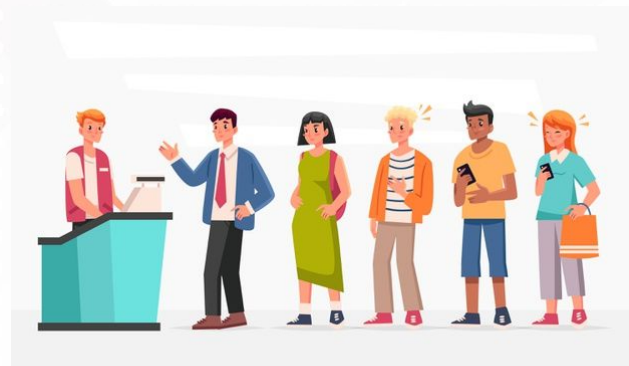
`meu_array.size`

- Podemos ordenar

`meu_array.sort()`

- Podemos inverter

`meu_array.reverse()`





# Obter um elemento do array

`meu_array[POSIÇÃO]`



# Como criar array no Kotlin?

Inicializando um array preenchido - `arrayOf(1, 2, 3)`

Inicializando um array vazio - `arrayOfNulls<TIPO>(TAMANHO)`

Inicializando um array de inteiro - `intArrayOf(1, 2, 3)`

Para cada tipo numérico - **`tipo`**`ArrayOf(ELEMENTOS)`

# Navegando num array



```
fun main() {  
    val animais = arrayOf("peixe", "batata", "camarão")  
  
    for (animal in animais) {  
        println(animal)  
    }  
}
```



# Navegando num array



```
fun main() {  
    val animais = arrayOf("peixe", "batata", "camarão")  
  
    animais.forEach { animal →  
        println(animal)  
    }  
}
```

The background features a light gray grid with a pattern of small white dots. In the top-right and bottom-left corners, there are clusters of colorful triangles (red, blue, and white) arranged in a geometric, pixelated style.

# Array tem tamanho fixo!

# Podemos criar um “array dinâmico”

## ArrayList

```
ArrayList<TIPO>()
```

```
arrayListOf(ELEMENTOS)
```

# Operações com ArrayList



- Adicionando elemento

```
meu_arrayList.add(ELEMENTO)
```

- Remover elemento

```
meu_arrayList.remove(INDEX)
```

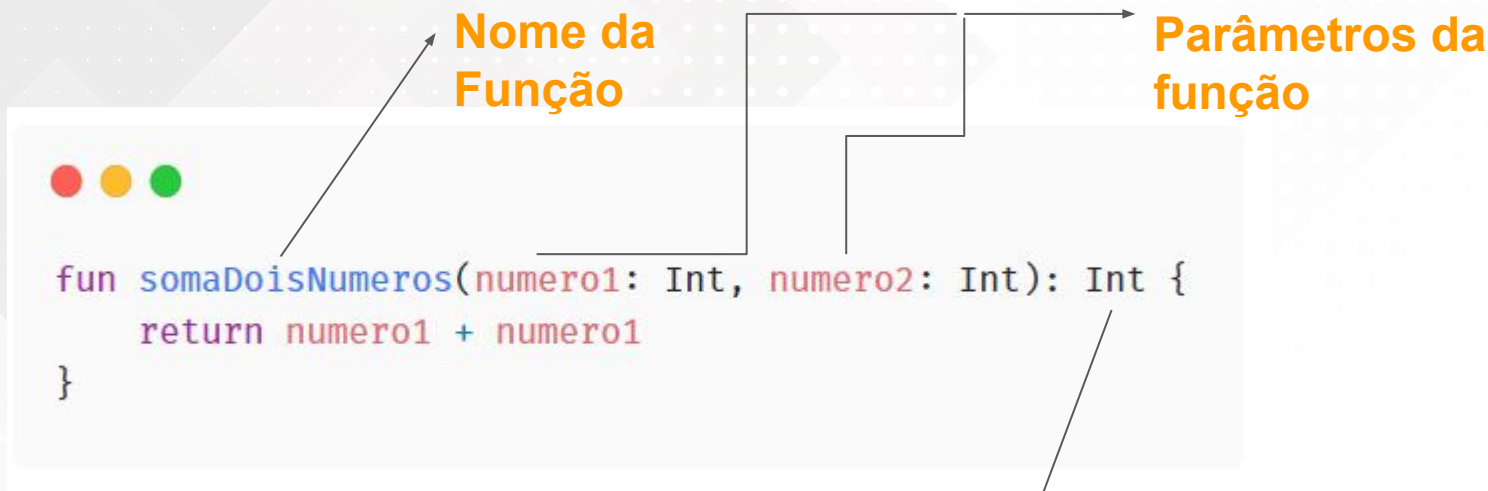
- Obter elemento

```
meu_arrayList[INDEX] ou meu_arrayList.get(INDEX)
```



# Prática I

# Função comum e retorno



Nome da Função

Parâmetros da função

```
fun somaDoisNumeros(numero1: Int, numero2: Int): Int {  
    return numero1 + numero1  
}
```

O tipo que ela tem que retornar



# Função comum e retorno



```
fun somaDoisNumeros(numero1: Int, numero2: Int) = numero1 + numero2
```



```
fun main() {  
    println(somaDoisNumeros(1, 2))  
}
```

```
fun somaDoisNumeros(numero1: Int, numero2: Int) = numero1 + numero2
```

# Let's Code!!

